1. In a database including primary and secondary servers and a replicator that copies database log entries between the primary server and the secondary server and replays said log entries on the secondary server, a method comprising:

initiating a transaction on the secondary server that includes reading a smart large object;

responsive to the initiating, creating a memory cache corresponding to the smart large object, said memory cache including a large object header; and

responsive to a replay on the secondary server of a log entry by said replicator that alters said smart large object, sending an exception to said transaction.

2. The method as set forth in claim 1, further comprising:

committing said replay of said log entry on the secondary server to alter said smart large object on the secondary server;

updating said large object header to generate an updated large object header; and

completing said transaction using said updated large object header.

3. The method as set forth in claim 2, wherein the sending of an exception to said transaction comprises:

sending an error code to a user read thread that generated said transaction.

4. The method as set forth in claim 1, further comprising:

committing said replay of said log entry on the secondary server to delete said smart large object on the secondary server; and

invalidating said large object header.

5. The method as set forth in claim **1**, further comprising:

allocating new space to said smart large object on the primary server, the allocating including:

allocating a memory page on the primary server to provide said new space, and

subsequent to the allocating of a memory page, updating a header of the smart large object on the primary server;

wherein the allocating of the memory page and the updating of the header are logged on the primary server to define at least a portion of said log entry log entry that alters said smart large object.

6. The method as set forth in claim **1**, further comprising:

deallocating memory from said smart large object on the primary server, the deallocating including:

updating a header of the smart large object on the primary server, and

subsequent to the updating, deallocating said memory;

wherein the updating and deallocating are logged on the primary server to define at least a portion of said log entry that alters said smart large object.

7. The method as set forth in claim 1, wherein the initiating of a transaction on the secondary server that includes reading a smart large object comprises:

initiating a transaction on the secondary server that includes reading one of a smart binary large object (smart-BLOB) and a smart character large object (smart-CLOB).

8. The method as set forth in claim 1, wherein the initiating of said transaction comprises:

initiating said transaction without locking the smart large object on the secondary server.

9. The method as set forth in claim 8, further comprising:

performing a modifying transaction on the primary server that alters said smart large object; and

logging said modifying transaction on the primary server to generate at least a portion of said log entry that alters said smart large object.

10. The method as set forth in claim 9, wherein the performing of said modifying transaction comprises:

acquiring a lock on said smart large object on the primary server.

11. In a database including primary and secondary servers and a high availability data replicator adapted to synchronize the secondary server with the primary

server, a smart large object application program interface residing on the secondary server, the smart large object application program interface comprising:

a smart large object read module adapted to read data from a smart large object without acquiring a lock on said smart large object, said read data being communicated to a client; and

an exception module adapted to send an exception to said client responsive to a synchronizing event of the high availability data replicator modifying said smart large object.

12. The smart large object application program interface as set forth in claim **11**, further comprising:

an update module adapted to update said smart large object responsive to said synchronizing event.

13. The smart large object application program interface as set forth in claim **12**, further comprising:

a cache module adapted to create a memory cache of said smart large object, said smart large object read module accessing said memory cache during said read, said cache module being further adapted to update said memory cache responsive to said updating of said large smart object.

14. An article of manufacture comprising one or more non-volatile storage media encoding instructions for performing a high availability data replication process that synchronizes a secondary server with a primary server of a database that includes

smart large objects, the process comprising:

ordering log entries of a smart large object modifying operation performed on the primary server in a selected order wherein a log entry corresponding to updating a large object header of said smart large object is consistent immediately upon execution;

transferring log entries including said log entries of said smart large object modifying operation from the primary server to the secondary server; and

replaying said transferred log entries on the secondary server, the replaying of said log entries of said smart large object modifying operation being performed in the selected order without locking said smart large object on the secondary.

15. The article of manufacture as set forth in claim **14**, wherein the process further comprises:

identifying a read operation accessing said smart large object; and

prior to the replaying of said log entries of said smart large object modifying operation, communicating an exception to a client associated with said read operation.

16. The article of manufacture as set forth in claim **15**, wherein:

the replaying of said log entries of said smart large object modifying operation includes modifying said smart large object on the secondary server; and

the communicating of an exception includes communicating an error code to said client.

17. The article of manufacture as set forth in claim **15**, wherein:

the replaying of said log entries of said smart large object modifying operation

includes deleting said smart large object on the secondary server; and

the communicating of an exception includes invalidating a large object header associated with said read operation.

18. The article of manufacture as set forth in claim **14**, wherein said smart large object modifying operation includes allocating memory to said smart large object, and said ordering comprises:

ordering a log entry corresponding to allocation of memory before a log entry corresponding to updating a header of the smart large object, whereby replaying of said log entries of said smart large object modifying operation on the secondary server in the selected order allocates memory on the secondary server before updating said header on the secondary server.

19. The article of manufacture as set forth in claim **14**, wherein said smart large object modifying operation includes deallocating memory from said smart large object, and said ordering comprises:

ordering a log entry corresponding to updating a header of the smart large object before a log entry corresponding to deallocation of memory, whereby replaying of said log entries of said smart large object modifying operation on the secondary server in the selected order updates said header on the secondary server before deallocating memory on the secondary server.

20. The article of manufacture as set forth in claim **14**, wherein said ordering of log entries of a smart large object modifying operation comprises:

ordering log entries of one of:

a smart binary large object (smart BLOB) modifying operation, and

a smart character large object (smart CLOB) modifying operation,

performed on the primary server in a selected order wherein a log entry corresponding to updating a large object header of said smart large object is consistent immediately upon execution.